Oracle White Paper
September 2013

The Department of Defense (DoD) and Open Source Software

ORACLE

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.

Introduction

With tight budgets, aggressive schedules, and a pressing need to adapt quickly to meet changing business conditions, organizations throughout the U.S. Department of Defense (DoD) see the appeal of the open source model.

Open source software (OSS) includes operating systems, applications, and programs in which the source code is published and made available to the public, enabling anyone to copy, modify and redistribute that code without paying royalties or fees. Open source “products” typically evolve through community cooperation among individual programmers as well as very large companies. An open source license permits anybody in the community to study, change and distribute the software for free and for any purpose.

At first glance it might seem that DoD organizations can avoid buying commercial software products simply by starting with open source software and developing their own applications. As we will see, total cost of ownership (TCO) for open source software often exceeds that of commercial software. While minimizing capital expenses by acquiring “free” open source software is appealing, the up-front cost of any software endeavor represents only a small fraction of the total outlay over the lifecycle of ownership and usage. And while cost effectiveness is important, it must be carefully weighed against mission-effectiveness.

This paper seeks to answer two questions:

1. What are the tangible and intangible costs that the government should bear under an open source licensing model?

2. What are the tradeoffs and risks associated with open source licensing models in relation to commercially available software?

Many people think of open source software as free software, but, as we shall see in this paper, this perspective does not consider the entire story. Oracle customers use commercial Oracle software products together with open source technologies in mission-critical environments to reduce costs, simplify manageability, address maintainability, improve availability, boost reliability, and improve performance and scalability.

One of the key questions to ask is this: who has to deal with the many “moving parts” of an enterprise solution? Consider the ripple effect of a software component upgrade. The software has to be re-tested within the program's systems. Anything that relies on that software also has to be re-tested since no open source project thoroughly tests all the potential combinations of frameworks. Commercial software vendors handle a great deal of this exacting work. They also document what has been tested and support the overall solution.

We will consider many other nuances of this problem in the pages that follow.

Software Costs in DoD Programs

The escalation of information-driven warfare has made the cost of new IT systems a major concern within the Department of Defense (DoD). Government organizations must create watertight business cases to gain executive buy-in and drive decisions about software infrastructure for individual programs. Carefully refining requirements to balance costs and expected innovations, project stakeholders continually strive to design strategies that minimize risks. A comprehensive business case should consider cost, schedule, performance and risk across the program lifecycle.

Lets begin by defining some terms.

- *Cost* is the total outlay required to operate the program from inception through deployment. It includes routine maintenance and upgrades as well as the comparative cost of decommissioning and replacement.
- *Schedule* is the program timetable, as measured against requirements milestones.
- *Performance* is the degree to which the program fulfills its stated requirements.
- *Risk* is the quantitative degree of probability to fail to meet the stated costs, schedule and/or performance requirements.
- *Open* is a reference to the degree to which the technology adheres to commercial technology standards, set by industry community organizations such as OASIS, OMG, W3C and the Java Community Process.
- *Proprietary* requires a more complex definition. It is often used to connote the opposite of *open,* meaning a solution that is *closed* through utilization of non-standard interfaces or lack of support for commercial standards. *Proprietary* is also used to differentiate software that is licensed for use by paying a license fee and for which only compiled binaries are available, versus software that has an inherent right to access the source code granted in an open source license. These two common usages are an improper conflation of concepts, since the licensing model for software has nothing to do with the degree of openness in its implementation. To explicitly differentiate these concepts, we will use the term *commercial* for software that requires a license fee for use, and reserve *proprietary* as an antonym for *open.*

> Note: It is important to recognize that many open source products actually have two versions—one that provides a "right to use" and one (usually far more capable) that requires a paid subscription or license for use. An example of this is RedHat Inc.'s JBoss Enterprise Suite, which requires a paid subscription for use. Failure to pay the subscription constitutes revocation of the "right to use" and is in fact commercial software. The truly open source version of JBoss, known as "WildFly," is available from JBoss.org, and is a different code base than the commercial version.

Government organizations must build business cases to address these considerations across the entire program lifecycle. The up-front cost of acquiring the software is only a small part of the overall expense. As organizations begin to realize this, they become better equipped to develop a business case that properly assesses and measures the impact of software infrastructure choices.

Acquisition officials in the Department of Defense should be focused on the greatest return on investment. In the case of software, this does not necessarily mean buying the cheapest product, even if the license and/or support costs appear negligible.

Another key issue facing government acquisition executives is the issue of indemnification. When a software company provides a product to its customers, it is expected to stand by its product, and is legally required to do so. In the case of open source, there are remarkably few companies willing or able to offer indemnification for their “products.” This is partly due to the nebulous issue of “ownership” in the open source community. Using open source software comes with potential risk associated with intellectual property rights claims that can arise at any point after said software has been incorporated into a solution, and could potentially result in royalties being due to the owner of the code in question.

Drilling Down into Software Costs

On average, hard costs—such as up front license fees, support contracts and hardware—account for 10 to 20 percent of the cost of a software project. The remainder includes “soft costs” such as facilities, power, cooling and labor for the development, testing, deployment, and maintenance of the system. To get a complete picture, it is advisable to also account for expenses associated with downtime, such as lost revenue, opportunities, and productivity as well as the need for extra IT resources or external professional services.

Labor is the largest cost factor in a software development project. These costs are impacted by system usability, training, continuity of qualified IT knowledge workers, quality of tooling and the need to modify or build required infrastructure functionality. Developer productivity, technical support, documentation, ease of software deployments, system administration, and system reliability all contribute to program costs. Even small improvements in these areas can have profound impacts on the total cost of a program. Testing is another big area of concern. While there are testing tools available within most open source software ecosystems, this is an area where commercial technologies far surpass open source alternatives. In licensing open source software as an alternative to commercial quality software, the government takes on the responsibility for testing and integrating the major software components.

Focusing on the easily identifiable and predictable hard costs, such as software licensing and annual support, can obscure the total lifecycle cost of a program. Technology

decisions based on short-term or up-front savings only consider 10 to 20 percent of total program costs. The best value to the government comes through optimizing developer productivity, providing reliable and scalable infrastructure, and reducing these soft costs.

The license costs associated with COTS products typically represent a small fraction of expenses across the entire program lifecycle. Lack of long-term planning causes many government organizations to under-allocate resources leading to budget shortfalls and a limited ability to introduce new capabilities.

Because labor is the largest portion of the costs of an IT project and COTS software is such a small component of the overall lifecycle cost of a program, government officials should carefully assess claims by IT services organizations that promote open source adoption over commercial alternatives. In many cases, the commercial alternatives lead the market because they have far superior capabilities to the open source projects. In those circumstances, adoption of open source has proven to actually increase the overall cost of the project, thereby resulting in program cost overruns. In many cases, these cost-overruns benefit the IT services organizations that advocated for the adoption of open source by resulting in greater services revenue even though the end customer does not receive the capabilities they need.

To obtain a complete picture of costs, government organizations should carefully analyze historical spending patterns and resource allocations across the expected life of each program. Through this analysis, organizations can make more informed decisions with respect to the processes, tooling, assets and people to support future programs. These decisions will serve as the foundation for a long-term organization and governance strategy.

Mission Assurance and Reliability of Software

Perhaps the most important issue in a major DoD system is reliability, which includes the ability to scale under heavy load as well as a system's security and information-assurance features. Testing and certification of an end-to-end solution can be extraordinarily expensive, especially if that system is changed frequently.

Load testing, system performance tuning, and system optimization are also expensive tasks. Commercial software companies have developed highly refined methodologies to perform these tasks. Don't underestimate the difficulties associated with testing open source software and incorporating required changes into the main development stream, especially when it comes to testing for robustness and reliability under load.

Should the Federal Government fund the task of testing and tuning open source software? Designing contractual terms and conditions that encompass every aspect of a system's potential performance is an extraordinarily difficult undertaking. If the main goal and perceived value of the program is delivering robust and timely capabilities to the warfighter, then the government should take a hard look at these issues.

Commercial software companies spend a substantial amount of time and money developing utilities for testing, designing and refining testing methodologies, conducting the actual testing, and ensuring end-to-end performance. For example, Oracle spends about $5 billion each year maintaining and upgrading its well-integrated, best-of-breed technology stack—orders of magnitude higher than similar investments made by open source vendors. Oracle thoroughly tests its products at an extremely high level of rigor as part of its basic business model. With this fact in mind, many DoD agencies use open source software to fill gaps in a solution, rather than as the entire solution. For mission-critical functionality, commercial software wins the day.

Mission Readiness: The Ultimate Test

In order to avoid compromising mission-readiness, software assets must be continually maintained, patched, and updated to keep them free of errors and compatible with the surrounding hardware and software environment. Bugs must be fixed immediately. Patches must be installed with regularity. New features often require a complete test suite, versus just a patch release. Program business is very careful about introducing new features, although they do it when it will solve an important-enough problem. No software product or application exists in isolation. Rather, enterprise software assets are part of an intricate web that includes the hardware platform, operating system, storage environment, database management system, development tools, management tools, and many types of middleware governing security, access control, portal integration, collaboration, business process management, social computing, mobile access, and many other functions.

Maintenance

Maintaining individual software assets within this dynamic enterprise environment is a huge job. It's one of the things that many government organizations pay COTS vendors to handle for them. However, if you have developed enterprise software systems in-house using open source software, the burden is on you to keep them accurate, up to date, performing well, and compatible with the surrounding IT environment. This can get tricky and even cause you to avoid necessary updates and bug fixes for fear of disrupting a homegrown system that has been painstakingly developed, tuned and tested.

Often these systems are in mission-critical production environments with high demand, and consequently cannot afford any downtime, so the risk of applying a patch needs to be as low as possible.

Finally, formal accreditation of a homegrown system is no simpler than commercial enterprise systems. The likelihood of a commercial system receiving previous accreditation within the DoD is exponentially larger than for a purely homegrown system.

Support

COTS vendors have detailed policies to address these concerns. For example, Oracle's industry-leading support organization delivers complete support for the entire technology stack—hardware platforms, database, middleware, applications, management tools, and the operating system itself. This enterprise-caliber support addresses one of the key requirements of DoD customers: keeping systems on-line and ready for action.

Standards

Adherence to standards is essential, whether in the context of open source or non-open source software. With today's open standards, customers are able to run both open source and commercial software side by side in production environments. Standards enable choice, and many open source "products" are a "one-of-a-kind" solution, which can be a risky proposition.

"Our goal in providing a complete suite of software is not to lock people into a given stack but to offer customers the benefit of having pre-integrated components," said Oracle Chief Corporate Architect Edward Screven in an interview with *Oracle Magazine*. "But customers aren't going to choose that stack if they feel that somehow they're locked into software only from Oracle. So it's very important for us to base our software on open standards."[1]

"Free" Software and Open Source Software

The "open source versus free software" debate brings the issue of licensing terms and conditions into sharp focus. If you are concerned with vendor lock-in, and you see open source solutions as a means of "owning" the code you want to run, a quick glance at the Open Source Initiative (OSI) can be daunting. Nearly 100 licensing regimes have gone through the license approval process. Open source is not free, nor is it easy to understand the strict legal terms and conditions associated with its use.

As agencies look for ways to cut development costs while reducing development time, the availability of open source components becomes more than just an attractive alternative. But simply downloading a publicly available component and including it in a project can lead to significant repercussions. "Failure to recognize these technical and legal implications can impact the long term usability of a developed product," notes John Dingman in *DoD Software Tech News*. "In the last three years of open source usage

[1] Schwerin, Rich, "Open for Business," *Oracle Magazine*, July/August 2001 (http://www.oracle.com/technetwork/issue-archive/2010/o40interview-086226.html).

throughout the DoD, there have been many lessons learned, one of them being that there are still challenges ahead."[2]

An Open Source Failure that Cost Billions and Impacted the Healthcare of America's Veterans

While open source software has proven its value in many DoD projects, experts differ on where OSS should end and COTS should begin. The public has been well acquainted with this controversy thanks to the government's well-publicized attempt to create an integrated electronic health record (iEHR) system for the Departments of Defense (DoD) and Veterans Affairs (VA), a project governed by the Interagency Program Office (IPO). Many industry observers cite the runaway costs and lengthy timeframes of this project as a good example of putting too much emphasis on open source software.[3]

Here's the back story: The Department of Defense and Veteran's Administration use separate medical databases that can neither translate nor communicate their data in a functional way. The National Defense Authorization Act of 2008 directed the departments to develop a single electronic health record system by 2009. They pushed that scheduled date of completion to 2017 after the plan hit a number of management, oversight, and planning snags.[4]

Five years and nearly $12 billion later, the IPO has little to show for its efforts. The soaring costs of this failed project have led both departments to rethink their approach to achieving EHR interoperability, two top VA officials told members of Congress.[5] Former VA Chief Information Officer Roger Baker, a long-time proponent of open source software, said the VA now plans to use an integrator to combine multiple pieces of commercial software—such as the pharmacy system—but not for the open source software packages.[6] Baker made a spirited defense of the VA's plan to save money and improve its own EHR system through an open source EHR improvement effort. That led to a revision of the iEHR plan, now estimated at $16 billion.[7]

---

[2] Dingman, John, "DoD and Open Source Software," *DoD Software Tech News*, February 2011.

[3] http://www.openhealthnews.com/tagged/interagency-program-office-ipo

[4] http://www.usnews.com/opinion/blogs/world-report/2013/04/04/department-of-defense-health-care-fails-veterans

[5] https://veterans.house.gov/witness-testimony/the-honorable-roger-w-baker

[6] http://www.nextgov.com/defense/whats-brewin/2012/06/hey-how-about-iehr-systems-integrator/56098/

[7] http://www.modernhealthcare.com/article/20130301/NEWS/303019955/?template=printpicart

The VA continues to embrace open source components including Open VistA (for the EHR repository), Janus (for the portal), Mirth (as the enterprise service bus), and MySQL (for data storage and management).

In its response to the VA's request for proposal, Oracle's middleware team recommended open source in conjunction with a commercial SOA environment that could consume a wide variety of backend services—both from commercial and open source—solving these expensive integration problems once and for all. According to this considered response, a GUI solution that is tightly coupled with medical domain functionality should be avoided. A tightly coupled solution leads to one-off functional applications that are difficult to integrate with the enterprise environment. The users are forced to interact with multiple disjointed user interfaces to get to the functionality they require and this can lead to a severe loss of productivity. In addition, one-off solutions developed with open source software compromise on flexibility and agility in addressing changes to business requirements and in the long run lead to high maintenance costs.

Open Source tends to be used successfully in simple, low-risk projects. In contrast, iEHR is a massive mission critical project with great complexity and many risks. Given everything else that IPO management was focused on, wasting time and effort on adopting unproven open source instead of going with proven technologies appears to be a failure of risk management and program management.

Congress remains unimpressed with the progress. "We have spent hundreds of millions of dollars," said Rep. Mike Michaud (D-Maine), ranking member of the Veterans Affairs Committee. "Delaying the delivery of an integrated—that is integrated, not interoperable—information-sharing system runs directly against Congressional intent, and ultimately hurts our veterans."[8]

Congress has now acted by instructing VHA to ensure that the iEHR should adhere to open architecture standards, including "modular designs based on standards with loose coupling and high cohesion that allows for independent acquisition of system components,"[9] agreed-upon technical standards, enterprise investment strategies that leverage proven system designs and "aggressive life-cycle sustainment planning."

Finally, the bill says emphatically: "By the point of full deployment decision, such a record must be at a generation 3 level or better for a health information technology system," which implies COTS. To say that the vendor community feels vindicated—albeit at the cost of years of frustration, largely wasted efforts and tens-of-millions of dollars of taxpayer funds—would be an understatement.

[8] http://fcw.com/articles/2013/07/11/va-dod-health-record-hearing.aspx

[9] http://www.gpo.gov/fdsys/pkg/BILLS-113hr1960pcs/pdf/BILLS-113hr1960pcs.pdf

Vendor Contributions to Open Source

Which brings us to our central topic: What do vendors bring to open source implementations, and how can the DoD utilize open source software to streamline implementations rather than derail them?

While advocating for the elite capabilities of its commercial technology stack, Oracle supports open source from four perspectives:

1. Oracle leverages open source within its Enterprise Software offerings to improve the stability, maintainability, and time to market for software products across Fusion Middleware and Oracle Applications.
2. Oracle invests in open source by actively sponsoring Open Source Community Projects and participates in a number of open source steering committees, along with other vendors, to insure customers have open and interoperable enterprise solutions.
3. Oracle provides open source capabilities at zero license cost to support capabilities such as infrastructure, virtualization, database, application server platform and the entire software development lifecycle. Oracle offers open source with support options to provide customers with cost-effective alternatives to build systems that are focused on open standards, small workloads, and standalone OA&M requirements. Oracle envisions customers will graduate to enterprise capabilities as the load, functional requirements, and deployment architecture grow more complex.
4. Oracle licenses commercial software to fill the gaps where open source does not fully address enterprise capability requirements. These include, but are not limited to, the following examples:
    - Automated Deployment & Provisioning
    - Extensible Operations, Administration, & Management
    - Enterprise-wide Monitoring & Troubleshooting
    - Extreme Performance, High Availability, & Scalability

Remember, open source simply refers to the licensing terms and conditions associated with the software, not the cost of the code itself. Major software vendors—both those selling commercial products and those marketing themselves as open source vendors—are acutely aware of this reality.

When problems occur in a large or complex software environment, it's often impossible to reproduce such occurrences with very simple test cases. DoD customers need a vendor that understands their complete environment and has the expertise to diagnose and resolve problems by drawing from their knowledge of and familiarity with the entire framework, as opposed to requesting a simple reproducible test case.

For example, Oracle has made significant investments in developing, testing, optimizing, and supporting open source software technologies such as MySQL, GlassFish, Java (OpenJDK), Linux, PHP, Apache, Eclipse, EclipseLink, Berkeley DB, NetBeans, VirtualBox, and Xen. For many years Oracle has pioneered these open source solutions as a viable choice for development and deployment.

Oracle delivers business-class support for its open source software, providing the availability and assurance for virtually all application software stacks as required by production environments. We want customers to have choices, and there are many situations in which these open source products can complement or extend a robust commercial solution.

Oracle is uniquely positioned and qualified to deliver and support open source software by eliminating risk through supporting the binaries from open source projects. In addition, Oracle implements rigorous methodology and proven processes to ensure that the open source software meets or exceeds specifications by subjecting it to the same standards, quality assurance, and interoperability testing as Oracle's commercial software.

"Oracle doesn't really have an open source-specific strategy," said Oracle Chief Corporate Architect Edward Screven. "What we have is an overall company strategy: to deliver complete, open, integrated solutions to our customers. This includes stacks of software and hardware that are built together and tested together and serviced together. And open source is part of that."[10]

As Screven goes on to explain, open source is both a development methodology and a business model for software, which is different from conventional software and conventional hardware but still a very effective way to build components that are valuable for customers. MySQL is a good example. Oracle's support for this open source product includes keeping it well integrated with the rest of the Oracle stack. So, for example, it is manageable through Oracle Enterprise Manager, the data can be backed up and coordinated through Oracle Secure Backup, and customers can deliver auditing records into Oracle Audit Vault.

Similarly, Oracle Fusion Middleware is a microcosm reflecting Oracle's overall approach to open source. There are many open source components that are part of Oracle Fusion Middleware. For example, many parts of Java itself are open source. The underlying Oracle WebCenter infrastructure of Oracle Fusion Middleware is based on Apache. There are lots of Apache components that are built into Oracle Fusion Middleware besides the Web listener.

[10] Schwerin, *op. cit.*

The Rise of an Open Source Application Server

Consider the popular open source application server known as GlassFish. Oracle provides support services and additional features through a commercial offering called Oracle GlassFish Server that extends the capabilities of GlassFish Server Open Source Edition. Enhancements from GlassFish Community engineers, in collaboration with Oracle engineers, have transformed GlassFish Server Open Source Edition to include rich features such as high availability, interoperability with Microsoft Windows, strong integration with NetBeans and Eclipse, and alignment with Oracle Solaris, OpenSolaris, and MySQL. Oracle GlassFish Server is tested with these and other open source products while also being integrated/aligned with other commercial products such as Oracle WebLogic Server. Oracle Coherence, CohWeb, and Enterprise Management packs for WebLogic also include GlassFish as an option.

Like all Oracle open source software products, Oracle GlassFish Server represents a less expensive, more flexible alternative to proprietary software, while still delivering powerful features and capabilities found in enterprise-class commercial products. GlassFish Server Open Source Edition can be downloaded and used at no cost. Thus, organizations can start using and deploying with no upfront costs. As needs grow, in terms of number of applications, implementation complexity, and criticality of support, organizations can switch to the commercial offering from Oracle.

Case in Point: A Global Combat Support System

For twenty-first century war-fighters, the forward edge of combat can be anywhere and everywhere, and their technology must be able to deploy with them, regardless of location or circumstance. Logistics can mean life or death for over 200,000 marines, motivating the Marine Corps to deploy Global Combat Support System—Marine Corps/Logistics Chain Management (GCSS-MC/LCM)—an integrated suite of Oracle solutions that supports the Corps' essential supply and maintenance system.

Based on Oracle Applications and Oracle Fusion Middleware technology, the Global Combat Support System connects Marine units in the field with supporting units to more effectively track, transport and deliver support to operating forces, both deployed and in garrison. The U.S. Marine Corps worked with Oracle Consulting to solve a multitude of critical problems:

- Requests for supply were either not fulfilled or fulfilled multiple times because of invisibility of order status
- Supply pipelines were clogged with redundant orders
- Readiness dropped from 100% to 50% in weeks because transitional status was not available
- Units were dependent on forward-positioned stocks dragging massive amounts of inventory to compensate

- Documentation on repairs and maintenance of weapons, vehicles, and other equipment was not timely or reliable
- Commanders lacked confidence in their logistics support

To help remedy these communications problems, Oracle implemented a custom solution that solves a common challenge: moving data from classified warfighter networks across relatively low-bandwidth networks to the sensitive but unclassified networks where most of the interaction with suppliers occurs.

The U.S. Marine Corps implemented the Oracle E-Business Suite, including Oracle Financial Management, Oracle Purchasing, Oracle Mobile Field Service and Oracle Supply Chain Planning. In addition to its Oracle E-Business Suite applications, the U.S. Marine Corps deployed Oracle Fusion Middleware, Oracle WebLogic Server Management Pack, Oracle XML Gateway, Oracle Business Process Management, Oracle Business Activity Monitoring, Oracle Web Services Manager and Oracle Advanced Queuing via a Federated Private Cloud.

Today, the Marine Corps has a mission-critical system that scales globally and that can handle disconnected, interrupted, low bandwidth, and high latency operations. This solution—built with commercial software—was the only way to ensure global scalability and performance for a secure, mission critical system that must be able to handle breaks in connectivity, loss of network service, low bandwidth communications and transactions that are characterized by long response times.

Conclusion: The Proper Use of Open Source

Today, just about every commercial software vendor leverages open source software. Many of these vendors are the primary contributors to major open source projects. In fact, it is precisely their level of expertise and rigor that has helped make open source what it is today. The skill required to successfully and economically blend source code into a commercially viable product is relatively scarce. It should not be done directly at government expense.

Government-sponsored community development approaches to software creation lack the financial incentives of commercial companies to produce low-defect, well-documented code and are not subject to the same market pressure at the software code level. Oracle helps ensure that open source software fits well within the surrounding infrastructure and provides a route to enterprise grade production. However, for the intensive, mission-critical capabilities required by most DoD projects, Oracle recommends its flagship commercial software products.

Dr. David A. Wheeler of *DoD Software Tech News* helps with a reality check: nearly all publicly available open source software is commercial software. “Unfortunately, many government officials and contractors fail to understand this,” he writes. “This misunderstanding can result in higher costs, longer delivery times, and reduced quality for

government systems. There are also legal risks: government officials and contractors who do not understand this, yet influence the selection or use of software, will probably fail to comply with U.S. law and regulations on commercial software."[11]

Ultimately, Oracle believes we should listen to our customers. We realize "open source" is a highly overloaded term used by people in various roles.

- A developer might say open source is "a development framework that I can download and use in my application development project."
- An architect might say open source is "software that adheres to open standards for interoperability."
- Invariably, every one of the individuals in these roles, as well as the most important role within the organization, the IT Business Owner, will believe that open source is "software that is free."

Oracle is one of the biggest proponents and contributors to open source within the industry. However, we understand that there is a proper time and place for this type of development. Hopefully this paper has helped clarify and enrich an understanding of the tenets of open source and how Oracle heavily invests in each of these viewpoints. While many conversations tend to focus on year-one licensing costs, such costs are only a small part of the overall outlay for any enterprise software project.

Oracle is committed to offering choice, flexibility, and a lower cost of computing for end users. By investing significant resources in developing, testing, optimizing, and supporting open source technologies such as Linux, PHP, Apache, Eclipse, Berkeley DB, MySQL, and others, Oracle is embracing and offering open source solutions as a viable way to complete simple software projects and as an adjunct to the development and deployment of more complex projects that are based on commercial software.

Appendix: Oracle's Key Open Source Initiatives

Berkeley DB: Oracle Berkeley DB is a family of open source, embeddable databases that allows developers to incorporate within their applications a fast, scalable, transactional database engine with industrial grade reliability and availability; it is the most widely used open source database in the world with deployments estimated at more than 200 million.

Eclipse: Oracle is a strategic developer and board member of the Eclipse Foundation, contributing developers and leadership to three Eclipse projects: Dali JPA Tools, JavaServer Faces (JSF), and BPEL. Oracle Enterprise Pack for Eclipse provides tools

[11] Wheeler, David, *DoD Software Tech News*, op. cit.

that make it easier for Eclipse users to develop applications utilizing specific Oracle Fusion Middleware technologies and Oracle Database.

GlassFish: A lightweight, flexible, and open source application server, GlassFish is always the first compatible implementation of the latest Java EE platform specification. To learn about Java EE features, go to the GlassFish home page, https://glassfish.org/

Hudson: An extensible Continuous Integration Server that has became a popular alternative to CruiseControl and other open source build servers.

InnoDB: Created by Oracle subsidiary Innobase OY, InnoDB is the leading transactional storage engine for the popular MySQL open source database.

Java: Java programming language and computing platform powers state-of-the-art programs including utilities, games, and business applications. Java runs on more than 850 million personal computers and on billions of devices worldwide.

Java Platform, Micro Edition (Java ME): Developers in the mobile and embedded community drive the evolution and adoption of Java ME for mobile and embedded devices.

Linux: Oracle's technical contributions to Linux enhance and extend enterprise-class capabilities, and Oracle Linux Support delivers enterprise-quality support for Linux at a lower cost.

MySQL: MySQL is the world's most popular open source database for the Web.

NetBeans: NetBeans offers a free, open source Integrated Development Environment (IDE) for software developers, along with all the tools to create professional desktop, enterprise, web, and mobile applications with Java, C/C++, PHP, JavaScript, Groovy, and Ruby. NetBeans Platform is the world's only modular Swing application framework.

OpenJDK: OpenJDK is an open source implementation of the Java Platform, Standard Edition (Java SE) specifications and is licensed as free software under the GNU General Public License (GPL).

PHP: Oracle is committed to enabling open source scripting language PHP for the enterprise with Zend Server.

VirtualBox: VirtualBox is available under the open source GNU General Public License (GPL) and offers powerful x86 and AMD64/Intel64 based desktop virtualization.

Xen: Oracle contributes heavily to feature development of Xen mainline software and is a member of the Xen Advisory Board. Part of Oracle VM, next generation server virtualization software, includes the Xen hypervisor.

Open source Tooling Projects: Oracle contributes to several open source tooling projects, including Project Trinidad (ADF Faces), Eclipse, Spring, and more.

DoD and Open Source

September 2013
Author: David Baum

Contributing Authors:

Alexandra Huff, John Clingan, Peter Bostrom

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com

Oracle is committed to developing practices and products that help protect the environment

Hardware and Software, Engineered to Work Together